Title	CFP: Cooperative fast protection
Author(s)	Wu, B; Ho, PH; Yeung, KL; Tapolcai, J; Mouftah, HT
Citation	IEEE INFOCOM'09 mini-conference, Rio de Janeiro, Brazil, April 2009. In Journal Of Lightwave Technology, 2010, v. 28 n. 7, p. 1102-1113
Issued Date	2010
URL	http://hdl.handle.net/10722/139272
Rights	Creative Commons: Attribution 3.0 Hong Kong License

CFP: Cooperative Fast Protection

Bin Wu, *Member, IEEE*, Pin-Han Ho, Kwan L. Yeung, *Senior Member, IEEE*, János Tapolcai, *Member, IEEE*, and Hussein T. Mouftah, *Fellow, IEEE*

***Abstract*—We introduce a novel protection scheme, called cooperative fast protection (CFP), to fight against a single link failure in survivable(wavelength division multiplexing (WDM) mesh networks. CFP achieves capacity-efficient fast protection with features of node-autonomy and failure-independency. Though CFP organizes spare capacity into pre-cross-connected cycles, it differs from p-cycle by reusing the released working capacity of the disrupted lightpaths (i.e., stubs) in a cooperative manner, and utilizing both the released stubs and the spare capacity on the cycles to set up backup paths. This is achieved by allowing all failure-aware nodes to switch traffic upon a link failure, such that the disrupted lightpaths can be restored even if the end nodes of the failed link are not on the cycles. CFP also differs from FIPP (Failure Independent Path Protecting) p-cycle by reducing optical recovery time, and not requiring the cycles to pass through the source nodes of the protected lightpaths. By jointly optimizing both working and spare capacity placement, we formulate an ILP (Integer Linear Program) for CFP design without candidate cycle enumeration. Theoretical analysis and numerical results show that CFP significantly outperforms p-cycle based schemes by achieving faster optical recovery speed with much higher capacity efficiency. The performance gain is achieved at the expense of higher computation complexity, but without involving any additional signaling mechanism in the optical domain.**

***Index Terms*—CFP (cooperative fast protection), optical networks, p-cycle (preconfigured protection cycle), survivability.**

I. Introduction

WAVELENGTH-DIVISION MULTIPLEXING (WDM) technology allows hundreds of high-speed wavelength channels (each with a bandwidth of 40 Gbps or above) to be multiplexed onto a single fiber for parallel data transmission. This greatly improves the efficiency of data transmission in optical networks, and dramatically cuts down the network cost as well. On the other hand, optical networks are vulnerable to failures such as link failures caused by fiber-cuts [1], [2]. Due to the high-speed nature of WDM optical networks, even a very short service downtime can lead to a huge amount of data and revenue loss. So, it is critical to achieve fast optical recovery against a failure.

It is well known that p-cycle based schemes [3]–[5] (to be reviewed in Section II) achieve relatively high capacity efficiency, with a much faster optical recovery speed than other schemes such as SBPP (Shared Backup Path Protection [6]). In this paper, we define a *fast protection* scheme as a scheme where the optical recovery speed of the disrupted lightpaths is not impaired by the setup time of the backup paths. Under this definition, both link-based p-cycle [3] and FIPP p-cycle [5] are fast protection schemes (for link and path protection, respectively). As pointed out in [7], fast protection achieved by the p-cycle based schemes is due to the fact that the backup paths are fully pre-cross-connected instead of being set up in real time using some signaling. It is interesting to ask whether fully pre-cross-connected backup path is a necessary condition for achieving fast protection, and whether we can find a fast protection scheme with both faster optical recovery speed and higher capacity efficiency than those p-cycle based schemes.

In this paper, we propose a novel protection mechanism called cooperative fast protection (CFP) to answer the above questions. CFP is a fast protection scheme with a distinct feature of cooperative stub reuse, and it organizes the spare capacity into a set of pre-cross-connected cycles. Upon a link failure, each lightpath passing through the failed link is disrupted, and the downstream working capacity beyond the failure point (defined as a *stub*) is released. CFP reuses the stubs of the disrupted lightpaths in a cooperative manner, together with the pre-cross-connected spare capacity on the cycles, to set up the backup paths. Here, the term "cooperative" means that the stub released from one lightpath can be reused to set up the backup path for another lightpath, and the backup paths are set up by cooperatively utilizing both the pre-cross-connected spare capacity and the stubs. The key to achieve cooperative stub reuse is to allow all failure-aware nodes to carry out traffic switching against a link failure, which will be detailed in Section III. We notice that "stub release" has been considered in true path restoration (PR) [8], which is a failure-dependent restoration scheme. The backup path in PR reuses a part of the released capacity of its own working path. In contrast, CFP reuses the stubs in a cooperative manner among different lightpaths.

A backup path in CFP may not be fully pre-cross-connected. If a stub terminates at a failure-aware node (to be defined in Section III) on a cycle, the corresponding backup path can be set up in real time by allowing the failure-aware node to switch traffic, such that the stub can be connected to the pre-cross-connected spare capacity on the cycle to form the backup path. In CFP, the real time setup process of the backup paths does not impair the optical recovery speed of the disrupted lightpaths.

Manuscript received February 08, 2009; revised September 25, 2009. First published December 01, 2009; current version published March 10, 2010. This paper was presented in part at the IEEE INFOCOM'09 mini-conference, Rio de Janeiro, Brazil, April 2009.

B. Wu and P.-H. Ho are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1 (e-mail: b7wu@uwaterloo.ca, pinhan@bbcr.uwaterloo.ca).

K. L. Yeung is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: kyeung@eee.hku.hk).

J. Tapolcai is with the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary (e-mail: tapolcai@tmit.bme.hu).

H. T. Mouftah is with the School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada, K1N 6N5 (e-mail: mouftah@site.uottawa.ca).

Digital Object Identifier 10.1109/JLT.2009.2037525

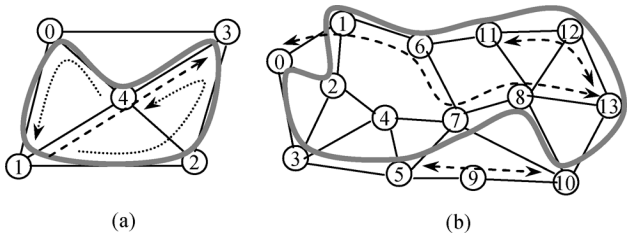

Fig. 1. p-cycle and FIPP p-cycle protection. (a) p-cycle. (b) FIPP p-cycle.

This qualifies CFP as a fast protection scheme as defined earlier. Besides, the switching activity at each failure-aware node is strictly failure independent and fully autonomous without requiring additional inter-node signalling. By making the best use of failure awareness of some nodes and enabling cooperative stub reuse, we show that CFP achieves faster optical recovery speed than link-based p-cycle, and higher capacity efficiency than FIPP p-cycle. Note that as a path protection scheme, FIPP p-cycle has higher capacity efficiency than link-based p-cycle but slower optical recovery speed.

The rest part of the paper is organized as follows. Section II reviews the p-cycle based fast protection schemes and analyzes their limitations. Section III presents the CFP mechanism based on extensive examples. An ILP (Integer Linear Program) is formulated in Section IV for CFP design to minimize the total working and spare capacity. Numerical results are presented in Section V, which demonstrate the superior performance and merits of CFP. We conclude in Section VI that CFP is a capacity-efficient fast protection scheme which significantly outperforms the existing counterparts.

II. p-Cycle Based Schemes and Their Limitations

A. Link-Based p-Cycle

Over the past decade preconfigured protection cycle (p-cycle) [3] has been considered as the most capacity-efficient link protection scheme that achieves the fastest optical recovery speed. As shown in Fig. 1(a), a p-cycle is a pre-cross-connected optical loopback implemented by using one unit of spare capacity (i.e., one backup wavelength) on each link it traverses. A link traversed by a particular p-cycle is called an *on-cycle link* of this p-cycle. If a link is not traversed by the p-cycle but its both end nodes are, then this link is called a *straddling link*. In bidirectional WDM networks, a p-cycle can protect one unit of working capacity on each on-cycle link and two units on each straddling link [3], [9]–[11]. In Fig. 1(a), if on-cycle link (3, 4) fails, the remaining part of the p-cycle provides one backup path to protect one unit of working capacity on (3, 4); if straddling link (1, 4) fails, two backup paths are available as indicated by the two dotted arrows. In fact, p-cycle achieves high capacity efficiency by sharing the spare capacity to protect all the on-cycle and straddling links. Fast optical recovery can be achieved because only the two end nodes of the failed link carry out traffic switching which can be done in a very responsive manner, and the resultant backup paths are fully pre-cross-connected.

Despite its excellent performance, p-cycle has some intrinsic features that limit its capacity efficiency and optical recovery speed: 1) a p-cycle can only protect its on-cycle and straddling links, but not those links with at least one end node off the cycle; 2) as a consequence of 1), a p-cycle tends to be large in size such that it can traverse or straddle as many links as possible in order to achieve better capacity efficiency. This increases the length of the backup paths, which decreases the optical recovery speed and promotes optical signal impairment en route. Though the size of each p-cycle can be limited [9], [11], it implies more p-cycles required in a given network, which not only decreases the capacity efficiency but also complicates the network management; 3) each disrupted lightpath must be rerouted from the upstream end node of the failed link to the downstream one. The backup path could be very long and capacity-inefficient compared with the case where the traffic is directly rerouted to the destination; and 4) upon a particular link failure, the downstream released working capacity (i.e., the stub) of each disrupted lightpath must be reused by the same lightpath instead of by others. Let a lightpath traverses through $1 \rightarrow 4 \rightarrow 3$ in Fig. 1(a). If link (1, 4) fails, the lightpath must be rerouted to $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$, and then reuse its own stub $4 \rightarrow 3$ to reach the destination. The rerouted lightpath passes through link (3, 4) twice in opposite directions. In a fast protection scheme, due to the features of pre-cross-connection and spare capacity sharing, some backup paths may contain a loopback where the restored traffic departing from a node on the backup path loops back to the same node. Such a loopback is defined as a *backhaul* [7]. The backhaul problem decreases both the capacity efficiency and the optical recovery speed (due to the additional distance on the loopback travelled by the rerouted traffic).

B. FIPP (Failure Independent Path Protecting) p-Cycle

The p-cycle concept is also extended to path and segment protection [4], [5]. Fig. 1(b) shows an example of FIPP (Failure Independent Path Protecting) p-cycle [5], which assumes bidirectional lightpaths on the same route. If a link or node fails, the end nodes of a disrupted lightpath will detect the failure, and then switch the traffic onto the pre-cross-connected spare capacity on the FIPP p-cycle. As shown in Fig. 1(b), there are three types of relations between a lightpath and a FIPP p-cycle: pure straddling relationship ($5 \leftrightarrow 9 \leftrightarrow 10$), pure on-cycle relationship ($11 \leftrightarrow 12 \leftrightarrow 13$) and partially straddling/on-cycle relationship ($0 \leftrightarrow 1 \leftrightarrow 6 \leftrightarrow 7 \leftrightarrow 8 \leftrightarrow 13$). Protecting the first two types of lightpaths is similar to that in the link-based p-cycle scenario and is independent of the specific failure location on the lightpath, and only the two end nodes of the disrupted lightpath carry out failure detection and switching. For the partially straddling/on-cycle lightpath $0 \leftrightarrow 1 \leftrightarrow 6 \leftrightarrow 7 \leftrightarrow 8 \leftrightarrow 13$, the situation is more complex. It can be disrupted due to a failure at on-cycle link (1, 6) or (7, 8), or at another link or node on the lightpath. Therefore, the switching nodes (i.e., nodes 0 and 13) need to know whether the upper arm of the FIPP p-cycle (i.e., the upper part of the cycle between the two switching nodes) or the lower arm is disrupted or not, and then switch the traffic to a viable arm accordingly. In [5], this was taken as a trivial issue without violating the failure independent property. Specifically, it was explained in [5] that the switching nodes can detect not only the disruption of the lightpath, but also the direction from which the loss of light (LOL) or alarm indication signal (AIS) of the FIPP p-cycle arrives. Then, the spare capacity in the other

direction of the FIPP p-cycle, or the predefined default direction if no LOL or AIS is observed on the cycle, can be used to reroute the lightpath.

As a path protection scheme, FIPP p-cycle achieves much higher capacity efficiency than link-based p-cycle. Let the *length* of a cycle be the number of links it passes through. The length of FIPP p-cycles tends to be shorter than that of link-based p-cycles, because FIPP p-cycles do not need to straddle or pass through as many individual links as in the link-based p-cycle scenario. However, some intrinsic features of FIPP p-cycle also limit its performance: 1) due to the nature of path-based protection, the optical recovery speed is slower than that in the link-based p-cycle protection. Not only the upstream on-the-way traffic ahead of the failure point will be lost, but also the switching nodes need to wait for the arrival of failure indication signals (such as LOL) before they can switch; 2) a FIPP p-cycle cannot protect a lightpath with an end node off the cycle; 3) the downstream released stub of each disrupted lightpath is not reused at all; 4) to keep the failure independent property, the pre-cross-connected spare capacity on the FIPP p-cycle could be underutilized for a specific failure location. In Fig. 1(b), if lightpath $0 \leftrightarrow 1 \leftrightarrow 6 \leftrightarrow 7 \leftrightarrow 8 \leftrightarrow 13$ fails due to a failure at link (6, 7), the FIPP p-cycle can protect only one unit of traffic, though there are two usable backup paths on the cycle; 5) to protect a partially straddling/on-cycle lightpath, the switching nodes need signals from both the disrupted lightpath and the FIPP p-cycle; and 6) FIPP p-cycle assumes bidirectional traffic on the same route. Otherwise it would be impossible for the source node of a directed lightpath to detect the failure by receiving a loss of light (LOL) indication in optical domain.

III. Cooperative Fast Protection (CFP)

Motivated by the above observations in both link-based and FIPP p-cycle scenarios, in this section we propose cooperative fast protection (CFP) to protect each lightpath against any single link failure in a directed WDM network. We first introduce the CFP mechanism and show how it works in a node-autonomous and failure-independent manner. Then, we analyze how fast protection can be achieved in a capacity-efficient way.

A. Definition of Failure-Aware Nodes

We observe that a link failure can be detected not only by the two end nodes of the failed link (as in the link-based p-cycle scenario), but also by the destination nodes of all disrupted lightpaths (as in the FIPP p-cycle scenario). Upon a link failure, the two end nodes of the failed link can detect the failure by various means such as observing fiber dark or loss of optical supervisory channel (OSC) [12]) signal, and the failure is defined as an *adjacent failure* of the two end nodes. On the other hand, all the lightpaths passing through the failed link are disrupted. If the failed link is not incident on the destination node of a disrupted lightpath, the destination node can detect this *remote failure* by a loss of light (LOL) indication on the lightpath (although it cannot accurately localize the failure). In CFP, the two end nodes of the failed link and the destination nodes of all the disrupted lightpaths are identified as *failure-aware nodes*. Due to the transparency of the network, we assume that other nodes in the network cannot sense the failure. As a unique feature of CFP, all those failure-aware nodes can initiate protection switching against the link failure without additional inter-node signalling. This has never been investigated in previous studies.

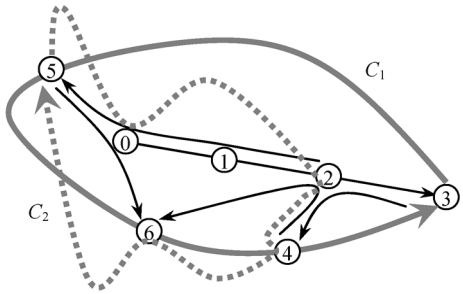

Fig. 2. Cooperative fast protection (CFP).

B. Working Principles of CFP

Similar to link-based and FIPP p-cycles, CFP organizes the spare capacity into pre-cross-connected protection cycles. The difference is that CFP allows stub reuse and enables more (failure-aware) nodes to switch the disrupted traffic. We use the example in Fig. 2 to illustrate how CFP works.

In Fig. 2, a failure at link (0, 1) can be detected as an adjacent failure by nodes 0 and 1, and as a remote failure by nodes 3 and 5 due to the disruption of lightpaths $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ and $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$. Meanwhile, the working capacity $1 \rightarrow 2 \rightarrow 3$ on the first lightpath and $0 \rightarrow 5$ on the second are released as stubs. Because the two lightpaths pass through (0, 1) in opposite directions, the stub released from one lightpath can be reused by the other. By utilizing both the stubs and the spare capacity on the solid (directed) cycle C_1, the backup path for $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ is $0 \rightarrow 5 \rightarrow 6 \rightarrow 4 \rightarrow 3$, and that for $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ is $2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 5$. Nodes 0 and 1 detect the adjacent failure and perform local switching to support the setup of the backup paths. Though nodes 3 and 5 cannot exactly localize the remote failure, they are still failure-aware and thus can properly connect the stubs to the spare capacity on C_1. For simplicity, if a lightpath can be protected against all possible link failures using the spare capacity on a cycle, we say that it can be protected by this cycle, although the protection may be assisted by some stubs. In CFP, each lightpath is protected by a single cycle, and each cycle can protect only those lightpaths with an on-cycle destination.

Upon a particular link failure, if the backup path of a lightpath l_1 reuses the stub of another lightpath l_2, we call l_2 the *partner* of l_1 at the failed link, where l_2 must pass through this link in the opposite direction of l_1 and its destination must be on the protecting cycle of l_1. However, l_1 may not be the partner of l_2 at the same time. In Fig. 2, lightpaths $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ and $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ are also partners of each other at link (1, 2). Therefore, if link (1, 2) fails, the two lightpaths can be protected in a similar way, but the set of failure-aware switching nodes is $\{1, 2, 3, 5\}$. Note that the switching activity at nodes 3 and 5 is independent of the failure location. No matter whether link (0, 1) or (1, 2) fails, nodes 3 and 5 carry out the same switching. They always receive the restored traffic from the viable arm on C_1, and connect the corresponding stub to the other arm of the cycle.

For example, if node 3 detects a remote failure on $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$, it receives the restored traffic from the on-cycle spare capacity $4 \rightarrow 3$ of C_1, and connects the stub of this lightpath to the on-cycle spare capacity $3 \rightarrow 5$. However, the situation is slightly different if the failure is adjacent to the destination of the lightpath. For example, if link (2, 3) fails, lightpath $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ is rerouted to $0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 3$, because it has a partner $3 \rightarrow 2 \rightarrow 4$ at link (2, 3), and nodes $\{2, 3, 4\}$ are failure-aware to make the proper switching. Node 3 receives the restored traffic from $4 \rightarrow 3$ on C_1 and switches the disrupted traffic of $3 \rightarrow 2 \rightarrow 4$ (instead of any stub) onto $3 \rightarrow 5$, where the backup path for $3 \rightarrow 2 \rightarrow 4$ is $3 \rightarrow 5 \rightarrow 6 \rightarrow 4$. Since node 3 detects an adjacent failure, it does not connect any stub to the spare capacity on C_1. On the other hand, the switching activity at node 3 is still failure-independent against any remote failure.

The above example shows how lightpath $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ is protected by C_1 against each possible link failure on the lightpath, though both its source node 0 and at least one end node of the failed link are not on C_1. The key points are: 1) a partner of $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ exists at every link along the lightpath, which provides a stub to bridge the disrupted traffic onto C_1; 2) all the failure-aware nodes can properly switch to set up the desired backup paths; and 3) the protection is node-autonomous and failure-independent, where each failure-aware node responds to a failure based on the locally observed OSC and LOL signals.

In addition to lightpath $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$, both $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ and $3 \rightarrow 2 \rightarrow 4$ in Fig. 2 can be protected by the solid cycle C_1, because each of them has a partner at every link along the lightpath. For example, lightpath $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ passes through three links (0, 1), (1, 2) and (0, 5). At the first two links, its partner is $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$. At link (0, 5), its partner is another lightpath $5 \rightarrow 0 \rightarrow 6$. However, lightpath $5 \rightarrow 0 \rightarrow 6$ cannot be protected by C_1. To keep the failure-independency feature in CFP, each lightpath must be protected by a single cycle, and its destination node must respond identically to any possible remote failure on the lightpath. Although a backup path $5 \rightarrow 6$ on C_1 can be found for $5 \rightarrow 0 \rightarrow 6$ against a specific failure at link (0, 5), the lightpath cannot be protected by C_1 against another failure at link (0, 6) due to the lack of a partner. In fact, lightpath $5 \rightarrow 0 \rightarrow 6$ is protected by the dotted cycle C_2 against the two possible link failures at (0, 5) and (0, 6). If either link fails, its upstream end node switches the traffic onto the spare capacity on C_2, whereas the destination node 6 of the lightpath always receives the restored traffic from the spare capacity $4 \rightarrow 6$ on C_2. Consider a failure at link (0, 5) where two lightpaths $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ and $5 \rightarrow 0 \rightarrow 6$ are disrupted. The set of failure-aware nodes is $\{0, 5, 6\}$. Nodes 0 and 5 detect an adjacent failure but node 6 detects a remote one. The switching at node 0 allows the backup path of $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ to reuse the stub $0 \rightarrow 6$ of $5 \rightarrow 0 \rightarrow 6$. Node 6 connects the stub $0 \rightarrow 6$ to C_1 but receives the restored traffic of $5 \rightarrow 0 \rightarrow 6$ from C_2. Meanwhile, node 5 switches the disrupted lightpath $5 \rightarrow 0 \rightarrow 6$ onto C_2 but receives the restored traffic of $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ from C_1. Accordingly, the backup path for $5 \rightarrow 0 \rightarrow 6$ is $5 \rightarrow 0 \rightarrow 2 \rightarrow 4 \rightarrow 6$ on C_2, and that for $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ is $2 \rightarrow 1 \rightarrow 0 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 5$ by utilizing the stub $0 \rightarrow 6$ and the spare capacity $6 \rightarrow 4 \rightarrow 3 \rightarrow 5$ on C_1. This example shows how the failure-aware nodes, stubs and spare capacity on the cycles work in a cooperative manner to protect all the disrupted lightpaths. With similar analysis as above, it is easy to see that lightpath $3 \rightarrow 2 \rightarrow 4$ is protected by C_1 whereas $4 \rightarrow 2 \rightarrow 6$ is protected by C_2. Note that we consider directed pre-cross-connection of the spare capacity on the protection cycles. Lightpath $4 \rightarrow 2 \rightarrow 6$ can be protected by C_2 against a failure at link (2, 4) because it passes through the link in the opposite direction of C_2, whereas a cycle cannot protect a lightpath if both of them pass through some links in the same direction.

Fig. 3 summarizes the switching policies (i.e., switching activities) of each failure-aware node, as well as some key definitions in CFP. Note that we will give more discussions on policy II in Section III.E (for an on-cycle node), and policy III is ensured by our ILP as formulated in Section IV. The switching policies in Fig. 3 are pre-planned and performed autonomously at each node against any possible single link failure. Although CFP involves more switching nodes than both link-based and FIPP p-cycles, it increases neither the hardware complexity of any node nor the network management cost due to the full autonomy of each failure-aware node.

C. Realization of Fast Protection

With stub reuse and traffic switching at all failure-aware nodes, CFP does not ensure fully pre-cross-connected backup paths, but it still achieves fast protection as explained below.

Again, let us take Fig. 2 as an example. If link (1, 2) fails, the backup path for lightpath $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ is $0 \rightarrow 1 \rightarrow 0 \rightarrow 5 \rightarrow 6 \rightarrow 4 \rightarrow 3$, which consists of the stub $1 \rightarrow 0 \rightarrow 5$ released from its partner $2 \rightarrow 1 \rightarrow 0 \rightarrow 5$ and the pre-cross-connected spare capacity $5 \rightarrow 6 \rightarrow 4 \rightarrow 3$ on C_1. Due to the transparency of the network, stub $1 \rightarrow 0 \rightarrow 5$ is all-optically connected and thus is equivalent to a pre-cross-connected path. Let the *reconfiguration time* of a node be the sum of the failure detection time (after an OSC or LOL indication arrives) and the time required for switching. Node 5 starts its reconfiguration slightly later than node 1, because the optical signal in stub $1 \rightarrow 0 \rightarrow 5$ needs time to be exhausted after the failure, and thus the arrival time of LOL at node 5 is slightly deferred due to the optical signal transmission in the stub. However, the restored traffic reuses the same stub and also experiences the same optical transmission delay to reach node 5. Assume that the reconfiguration time is the same at each node. Node 5 should have already completed its reconfiguration when the restored traffic arrives. So, the reconfiguration at node 5 is transparent (or invisible) to the restored traffic. As a result, the optical recovery speed is not impaired by the setup time of the backup path, and it can be as fast as in the link-based p-cycle scenario. Moreover, CFP may restore the traffic even faster than link-based p-cycle due to the following facts (verified by the numerical results in Section V): 1) the backup path in CFP directly connects to the destination of the lightpath along the cycle, and thus the backhaul problem [7] can be effectively suppressed; and 2) the protection cycles in CFP do not need to traverse or straddle as many links as in the link-based p-cycle protection, and thus they tend to have a much shorter cycle length.

Definitions and Switching Policies in CFP

Definitions:

Definition 1 (Cooperative Fast Protection): Cooperative Fast Protection (CFP) is a capacity-efficient fast protection scheme with a distinct feature of cooperative stub reuse. CFP achieves cooperative stub reuse by allowing all failure-aware nodes in the network to carry out traffic switching, where the switching at each failure-aware node is node-autonomous and failure-independent.

Definition 2 (Fast Protection): A fast protection scheme is a scheme where the optical recovery speed of the disrupted lightpaths is not impaired by the setup time of the backup paths.

Definition 3 (Stub): If a lightpath is disrupted upon a link failure, its downstream working capacity beyond the failure point, which is all-optically connected between the downstream end node of the failed link and the destination node of the lightpath, is defined as a stub.

Definition 4 (Cooperative Stub Reuse): Cooperative stub reuse is a distinct feature of CFP. It means that the stub of one disrupted lightpath can be reused to set up the backup path for another disrupted lightpath, and the backup paths are set up by cooperatively utilizing both the stubs and the pre-cross-connected spare capacity.

Definition 5 (Failure-Aware Nodes): Upon a link failure, all nodes that can sense the failure event in optical domain are defined as failure-aware nodes. In CFP, the set of failure-aware nodes includes the two end nodes of the failed link and the destination nodes of all disrupted lightpaths.

Definition 6 (Partner): Upon a link failure, if the backup path of a lightpath l_1 reuses the stub of another lightpath l_2, we call l_2 the partner of l_1 at the failed link, where l_2 must pass through this link in the opposite direction of l_1 and its destination must be on the protecting cycle of l_1. Note that l_1 may not be the partner of l_2 at the same time.

Switching Policies at Each Failure-Aware Node:

Policy I: If the destination node of a lightpath detects an adjacent or remote failure on the lightpath, it always receives the restored traffic from the cycle that can protect this lightpath;

Policy II: If a node detects an adjacent failure, each lightpath that is going to the failed link is switched onto the stub of its partner. If a lightpath has no partner at the failed link, then the disrupted traffic is switched onto the cycle that can protect this lightpath;

Policy III: If the destination node of a lightpath detects a remote failure and the lightpath is the partner of other lightpaths at different links, it connects the stub to a single cycle that can protect all those lightpaths.

Fig. 3. Key definitions and switching policies in CFP.

D. Backhaul Problem

The backhaul problem is common in link-based fast protection schemes, and is one of the causes of impairing both capacity efficiency and recovery speed. Compared with the link-based p-cycle protection, CFP can effectively suppress the backhaul problem, because the disrupted lightpath is rerouted directly to its destination node along the cycle, instead of the downstream end node of the failed link. If both the destination node of the disrupted lightpath and the downstream end node of the failed link are on the protecting cycle, in the link-based p-cycle scenario the restored traffic may pass through the former and be forwarded to the latter, and then loops back. This will never happen in CFP. Fig. 2 gives an example. If link (2, 4) fails, CFP reroutes $4 \rightarrow 2 \rightarrow 6$ directly to the spare capacity $4 \rightarrow 6$ on C_2. In contrast, the link-based p-cycle protection first reroutes it to $4 \rightarrow 6 \rightarrow 5 \rightarrow 0 \rightarrow 2$ along C_2 to reach the end node 2 of the failed link, and then reuses the stub $2 \rightarrow 6$ of itself to reach the destination node 6. This incurs a long detoured backhaul with a loopback to node 6.

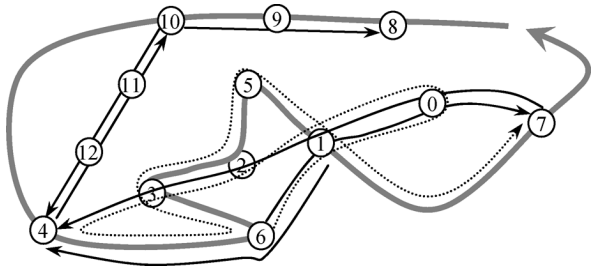

Fig. 4. Backhaul problem and capacity efficiency in CFP.

However, the backhaul problem is not totally removed from CFP, and it could happen when the stub is incident on the cycle multiple times. As illustrated in Fig. 4, lightpath $7 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ is a partner of $6 \rightarrow 1 \rightarrow 0 \rightarrow 7$ at links (0, 1) and (0, 7). If link (0, 7) fails, the end-to-end backup path for $6 \rightarrow 1 \rightarrow 0 \rightarrow 7$ is still subject to the backhaul problem, as shown by the dotted arrow in Fig. 4. Compared with a link-based p-cycle, a CFP cycle tends to be shorter in length and it does not need to straddle or pass through every link on the protected lightpath. As a result, a CFP cycle tends to avoid huge "concaves" (such as $6 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 1$ in Fig. 4), and thus long backhauls as in Fig. 4 are rarely observed.

E. Switching Priority of an On-Cycle Node

As summarized in the switching policy II in Fig. 3, if an on-cycle node detects an adjacent failure, it always switches the disrupted traffic onto the stub of the corresponding partner. At this on-cycle node, the traffic can be switched onto the spare-capacity on the cycle only if a partner cannot be found. But, this policy may lead to some redundant backhauls. Fig. 4 gives an example. If link (0, 1) fails, the backup path for $7 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ is $7 \rightarrow 0 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 4$. There are two possible backup paths for $6 \rightarrow 1 \rightarrow 0 \rightarrow 7$. The first one is $6 \rightarrow 1 \rightarrow 7$, where the traffic is directly switched onto the cycle at node 1, and it shares the spare capacity on the cycle with $7 \rightarrow 0 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 4$ in a conflict-free manner. Although node 4 still connects the stub $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ to the cycle upon detecting a remote failure, it does not matter because both the stub and the spare capacity $4 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 1$ on the cycle are not utilized in this case. In the second backup path, the traffic is switched onto the stub $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ at node 1, and then is forwarded onto the cycle at node 4 before it goes to the destination node 7 along the cycle. The second backup path includes a redundant backhaul because the traffic loops back to node 1, but it is chosen in CFP according to the switching policy II as summarized in Fig. 3.

The reason for the above choice is that disobeying the switching policy II may lead to a conflict in utilizing the spare capacity on the cycle. Assume that an on-cycle node is not the destination node of a disrupted lightpath, and upon detecting an adjacent failure it switches the traffic onto the cycle instead of the stub. Under this assumption (which violates the switching policy II), if link (1, 6) in Fig. 4 fails, the backup path for $6 \rightarrow 1 \rightarrow 0 \rightarrow 7$ would be $6 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 7$, and that for $1 \rightarrow 6 \rightarrow 4$ would be $1 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 4$. Then, both backup paths need to utilize the spare capacity

on link (1, 7), which leads to a conflict. A possible way to avoid the conflict is to let node 6 switch the disrupted traffic of $6 \rightarrow 1 \rightarrow 0 \rightarrow 7$ onto the cycle, but node 1 switch the disrupted traffic of $1 \rightarrow 6 \rightarrow 4$ onto the stub $1 \rightarrow 0 \rightarrow 7$ before it is forwarded onto the cycle at node 7. However, it is generally difficult to figure out such discrepancies among different switching nodes. By always switching the traffic onto the stub (if there is one), we can keep the same switching policy at every node to construct a CFP solution without any conflict in spare capacity utilization. Based on the solution obtained, any redundant backhauls can be easily identified and simply removed by slightly modifying the switching activities at the corresponding on-cycle nodes (if we want to further refine the CFP solution for a faster optical recovery speed).

F. Capacity Efficiency

Due to the feature of cooperative stub reuse, CFP is more capacity-efficient than the link-based p-cycle protection, because it can fight against a link failure even if the end nodes of the failed link are not on the protection cycle (e.g., a failure at link (11, 12) in Fig. 4). The fact that CFP suffers less from the backhaul problem also supports its higher capacity efficiency than link-based p-cycle. The superiority of CFP over link-based p-cycle will be further demonstrated by our numerical results in Section V. In fact, the capacity efficiency of CFP is even higher than that of the path-based FIPP p-cycle protection. Those pure straddling lightpaths (such as $4 \leftrightarrow 12 \leftrightarrow 11 \leftrightarrow 10$ in Fig. 4) and pure on-cycle lightpaths (such as $10 \rightarrow 9 \rightarrow 8$ in Fig. 4), which are protected in the FIPP p-cycle scenario, can also be protected in CFP but under a different mechanism with a faster optical recovery speed. If a partially straddling/on-cycle lightpath is disrupted due to any failure on the lightpath, its protecting FIPP p-cycle can protect only one unit of traffic for this lightpath. As we can see in Fig. 4, if link (0, 1) fails, the CFP cycle protects not only the partially straddling/on-cycle lightpath $7 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$, but also another lightpath $6 \rightarrow 1 \rightarrow 0 \rightarrow 7$. This gives CFP higher capacity efficiency than FIPP p-cycle. Besides, FIPP p-cycle assumes bidirectional lightpaths on the same route. CFP removes this assumption and thus is more general for a mesh WDM network. By taking each bidirectional lightpath as two separate directed lightpaths in opposite directions, CFP can be applied to protect bidirectional lightpaths (such as $4 \leftrightarrow 12 \leftrightarrow 11 \leftrightarrow 10$ in Fig. 4) as well.

IV. ILP Formulation

A. General Idea

We consider a joint design by optimizing the allocation of both working and spare capacity under a given traffic matrix. In addition to the objective function on minimizing the total capacity required, the ILP organizes its constraints into three parts: cycle formulation, routing and protection.

Cycle formulation is based on a recently proposed Cycle Exclusion technique [11], [13]. Since we do not know the exact number of protection cycles required until a solution is obtained, a constant J is defined as the maximum number of cycles allowed in the solution. If J is set large enough and the ILP returns less than J cycles, then the optimality of the solution can be ensured. Each cycle $C_j (1 \leq j \leq J)$ consists of a set of on-cycle *vectors* as shown in Fig. 5. A vector denotes an on-cycle backup wavelength (i.e., spare capacity) with a proper optical transmission direction. To formulate cycles, we can require each node in the network to have either a pair of inbound and outbound vectors, or no vector incident on it. But this may result in multiple disjoint cycles without traversing any common node and link (as illustrated in Fig. 5). Hence, the Cycle Exclusion technique is proposed [11], [13] to ensure a single cycle in formulating each C_j. The key idea is to assign a *voltage* value to each vector, and the voltage values must keep increasing along the cycle. In other words, the outbound vector of a node must have a larger voltage than its inbound vector. At the same time, a unique *reversal node* is defined in formulating each C_j, which is the only node that can have a smaller voltage on its outbound vector (than that on the inbound vector). This is called the *voltage constraint*. In Fig. 5, the voltage values keep increasing along the solid cycle, but the voltage 0.01 of the outbound vector at the reversal node is smaller than 0.05 of its inbound vector. If multiple disjoint cycles exist, only the one passing through the unique reversal node can exist, and all other disjoint cycles will be excluded by violating the voltage constraint. As a result, a single cycle is ensured in C_j.

The routing part in the ILP is based on the flow conservation property [14] of each lighpath. Note that the traffic demand between two communicating nodes may require multiple units of working capacity. Each unit is treated as a distinct lightpath and is separately routed. The lightpath starts at its source and terminates at its destination, whereas all other nodes in the network must obey flow conservation for this lightpath.

The protection part formulates how each lighpath is protected against each possible link failure. In particular, a lightpath can be protected by a cycle only if its destination node is on this cycle. Upon a link failure, all the lightpaths passing through the failed link in both directions are disrupted and the stubs are released. If multiple cycles pass through the destination node of a disrupted lightpath, the stub of the lightpath can be connected to at most one cycle, which may not be the cycle that protects this lightpath (recall our earlier example in Fig. 2). Define the lightpaths passing through the failed link in the same direction as *peers*. Among all the peers, at most one can have its stub connected to a particular cycle. As a result, a cycle can simultaneously protect a lightpath and its partner, but not two peers. To keep the feature of failure-independency, the stubs resulting from different link failures on a lightpath must be connected to the same cycle (this requirement is defined as the *consistency constraint*). Besides, if a lightpath l is protected by a cycle C_j but its stub is connected to another cycle, then the stub of any other peer of l cannot be connected to C_j (this requirement is defined as the *sovereignty constraint*). Consider lightpath $5 \rightarrow 0 \rightarrow 6$ in Fig. 2 which is protected by the dotted cycle C_2. If link (0, 5) fails, the backup path is $5 \rightarrow 0 \rightarrow 2 \rightarrow 4 \rightarrow 6$ on C_2, and the stub $0 \rightarrow 6$ is connected to the solid cycle C_1 at node 6. Suppose there is a peer $5 \rightarrow 0 \rightarrow 1 \rightarrow 2$ across the failed link (0, 5) with its stub $0 \rightarrow 1 \rightarrow 2$ connected to C_2 at node 2. Then, the sovereignty constraint is violated. Due to the switching activity at node 2, the disrupted lightpath

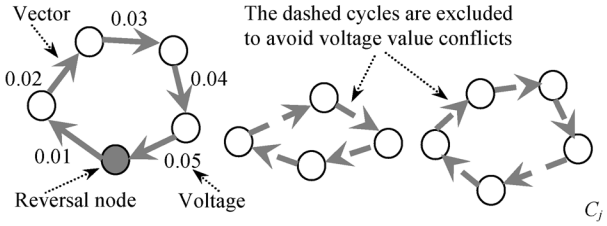

Fig. 5. Cycle exclusion (other nodes and links in the network are omitted).

$5 \rightarrow 0 \rightarrow 6$ cannot be properly restored using its backup path $5 \rightarrow 0 \rightarrow 2 \rightarrow 4 \rightarrow 6$.

B. ILP Formulation

The ILP for CFP design is formulated as follows. We first define the notations below, and then present the ILP with a brief explanation on each constraint.

Input Parameters:

- J The maximum number of protection cycles allowed in the CFP solution.
- j Protection cycle index where $j \in \{1, 2, \ldots, J\}$.
- $\boldsymbol{V}$ The set of all the nodes in the network.
- $\boldsymbol{E}$ The set of all the directed links in the network, where two directed links (u, v) and (v, u) pass through the same physical link in opposite directions.
- c_{uv} The cost of adding one unit of working or spare capacity to link (u, v) and $c_{uv} = c_{vu}$. If hop-count is used as the cost metric, then $c_{uv} = 1$ for each link (u, v). Otherwise c_{uv} may include distance-related cost.
- $\boldsymbol{L}$ A given traffic matrix. It denotes the set of all the lightpaths. An entry L_{sd} in $\boldsymbol{L}$ denotes L_{sd} distinct lightpaths between source s and destination d. For simplicity, we use $l \in \boldsymbol{L}$ to denote a lightpath l.
- λ A predefined positive fraction where $1/\|\boldsymbol{E}\| \geq \lambda > 0$. It is the minimum step that the voltage values increase along the vectors on each cycle.
- $s(l)$ The source node of lightpath l.
- $d(l)$ The destination node of lightpath l.
- C_L The length limit of each CFP cycle.
- W The number of available wavelengths on each fiber.

Decision Variables:

- e_{uv}^j Binary variable. It takes 1 if cycle C_j passes through link (u, v), and 0 otherwise.
- w_l^{uv} Binary variable. It takes 1 if lightpath l passes through link (u, v), and 0 otherwise.
- r_u^j Binary variable. It takes 1 if node u is the reversal node in formulating a cycle C_j, and 0 otherwise.
- z_u^j Binary variable. It takes 1 if cycle C_j passes through node u, and 0 otherwise.
- q_{uv}^j Fractional variable. It is the voltage value of the vector on link (u, v) in formulating cycle C_j. It takes 0 if there is no vector on (u, v).
- y_l^j Binary variable. It takes 1 if lightpath l can be protected by cycle C_j, and 0 otherwise.
- x_{luv}^j Binary variable. It takes 1 if the stub of lightpath l is connected to the spare capacity on cycle C_j upon a remote failure at link (u, v), and 0 otherwise.
- h_l^j Binary variable. It takes 1 if the stub of lightpath l is connected to the spare capacity on cycle C_j upon any remote failure on l, and 0 otherwise.
- p_{luv}^j Binary variable. It takes 1 if lightpath l passes through link (u, v), and is protected by cycle C_j. Otherwise it is 0.
- g_{uv}^j Binary variable. It takes 1 if cycle C_j passes through node u but C_j does not pass through link (u, v) from node u to node v, and 0 otherwise.

Objective:

$$\text{minimize} \left\{ \sum_j \sum_{(u,v)\in \boldsymbol{E}} c_{uv} e_{uv}^j + \sum_{l \in \boldsymbol{L}} \sum_{(u,v)\in \boldsymbol{E}} c_{uv} w_l^{uv} \right\}. \tag{1}$$

Objective (1) minimizes the total working and spare capacity.

Cycle Formulation Constraints:

$$\sum_{u \in \boldsymbol{V}} r_u^j \leq 1, \quad \forall j \tag{2}$$

$$e_{uv}^j + e_{vu}^j \leq 1, \quad \forall (u, v) \in \boldsymbol{E}, \quad \forall j \tag{3}$$

$$\sum_{(u,v)\in \boldsymbol{E}} e_{uv}^j = \sum_{(v,u)\in \boldsymbol{E}} e_{vu}^j, \quad \forall u \in \boldsymbol{V}, \quad \forall j \tag{4}$$

$$z_u^j = \sum_{(u,v)\in \boldsymbol{E}} e_{uv}^j, \quad \forall u \in \boldsymbol{V}, \quad \forall j \tag{5}$$

$$q_{uv}^j \leq e_{uv}^j, \quad \forall (u, v) \in \boldsymbol{E}, \quad \forall j \tag{6}$$

$$r_u^j + \sum_{(u,v)\in \boldsymbol{E}} q_{uv}^j - \sum_{(v,u)\in \boldsymbol{E}} q_{vu}^j \geq \lambda z_u^j, \quad \forall u \in \boldsymbol{V}, \quad \forall j. \tag{7}$$

The set of constraints (2)–(7) is for cycle formulation to ensure a single cycle in formulating each C_j. In particular, constraint (2) defines a unique reversal node. Constraint (3) requires each physical link to support at most one vector in either direction (but not both). Constraint (4) requires each node to have an equal number of inbound and outbound vectors incident on it. Constraint (5) identifies whether a node is traversed by the cycle or not, and it also confines each node to have at most one outbound vector. Constraint (6) enables a positive voltage value for each vector. The voltage constraint in (7) says that, if a node traversed by the cycle is not the reversal node, its outbound vector must have a larger voltage value than its inbound vector.

Routing Constraints:

$$w_l^{uv} + w_l^{vu} \leq 1, \quad \forall l \in \boldsymbol{L}, \quad \forall (u, v) \in \boldsymbol{E} \tag{8}$$

$$\sum_{(s(l),v)\in \boldsymbol{E}} w_l^{s(l)v} = 1, \quad \forall l \in \boldsymbol{L} \tag{9}$$

$$\sum_{(u,d(l))\in \boldsymbol{E}} w_l^{ud(l)} = 1, \quad \forall l \in \boldsymbol{L} \tag{10}$$

$$\sum_{(u,f)\in \boldsymbol{E}} w_l^{uf} = \sum_{(f,v)\in \boldsymbol{E}} w_l^{fv}, \quad \forall l \in \boldsymbol{L}, \quad \forall f \in \boldsymbol{V} : f \neq s(l), f \neq d(l). \tag{11}$$

The set of constraints (8)–(11) formulates the routing of each lightpath. Specifically, constraint (8) prevents the lightpath to pass through any link twice. Constraints (9)–(10) stipulate that each lightpath emanates at its source node and terminates at its destination node. Constraint (11) requires all other nodes to obey flow conservation.

Protection Constraints:

$$y_l^j \leq z_{d(l)}^j, \quad \forall l \in \boldsymbol{L}, \quad \forall j \tag{12}$$

$$x_{luv}^j \leq \frac{1}{2}\left(w_l^{uv} + z_{d(l)}^j\right), \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{13}$$

$$\sum_j x_{luv}^j \leq 1, \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E} \tag{14}$$

$$\sum_{l\in \boldsymbol{L}} x_{luv}^j \leq 1, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{15}$$

$$(1 - w_l^{uv}) + x_{luv}^j \geq h_l^j, \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{16}$$

$$h_l^j \geq x_{luv}^j, \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{17}$$

$$p_{luv}^j \leq \frac{1}{2}\left(y_l^j + w_l^{uv}\right), \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{18}$$

$$(1 - w_l^{uv}) + p_{luv}^j \geq y_l^j, \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{19}$$

$$\left(1 - p_{luv}^j\right) + h_l^j \geq \sum_{l\in \boldsymbol{L}} x_{luv}^j, \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{20}$$

$$\sum_{l\in \boldsymbol{L}} p_{luv}^j \leq 1, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{21}$$

$$y_l^j + w_l^{uv} + e_{uv}^j \leq 2, \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{22}$$

$$g_{uv}^j \leq \frac{1}{2}\left(z_u^j + (1 - e_{uv}^j)\right), \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{23}$$

$$y_l^j \leq (1 - w_l^{uv}) + \sum_{l\in \boldsymbol{L}} x_{lvu}^j + g_{uv}^j, \quad \forall l \in \boldsymbol{L}, \quad \forall (u,v) \in \boldsymbol{E}, \quad \forall j \tag{24}$$

$$\sum_j y_l^j = 1, \quad \forall l \in \boldsymbol{L}. \tag{25}$$

Protection constraints are formulated in (12)–(25). As indicated by (12), a lightpath can be protected by a cycle only if its destination node is on the cycle. By constraint (13), if the stub of lightpath l can be connected to cycle C_j upon a failure at link (u,v), then l must pass through (u,v) and its destination node must be on C_j. Constraint (14) requires the stub of each lightpath to be connected to at most one cycle. Constraint (15) means that, for any link failure, only one lightpath among all the peers can have its stub connected to a cycle. The consistency constraint is formulated in (16)–(17). According to (16), if lightpath l passes through link (u,v) but its stub is not connected to cycle C_j upon a failure at (u,v), then the stub resulting from any other link failure on l cannot be connected to C_j. Otherwise, the stub resulting from each possible link failure on l must be connected to the same cycle C_j, as formulated in (17). Constraints (18)–(19) define p_{luv}^j, which equals to 1 if lightpath l passes through link (u,v) and is protected by C_j. The sovereignty constraint is formulated in (20). If link (u,v) fails and the stub of a lightpath l protected by C_j is not connected to C_j, then the stub of any other peer of l cannot be connected to C_j. Constraint (21) means that a cycle cannot protect two or more peers against a link failure. Constraint (22) indicates that a cycle cannot protect a lightpath if both of them pass through any on-cycle link in the same direction. Constraint (23) defines g_{uv}^j, which equals to 1 if C_j passes through node u but not the directed link (u,v). However, $g_{uv}^j = 1$ does not prevent C_j to pass through (v,u) from v to u. Constraint (24) says that, if lightpath l passes through link (u,v) and it can be protected by C_j against a failure at (u,v), then it must find a partner at (u,v). Otherwise, g_{uv}^j must be 1 (i.e., the upstream end node u of the failed link must be on C_j, and lightpath l does not pass through (u,v) in the same direction as C_j). Finally, constraint (25) ensures that every lightpath is protected by a cycle.

Optional Constraints: If we have a length limit C_L on each CFP cycle, we can include the following constraint (26) in the ILP.

$$\sum_{(u,v)\in \boldsymbol{E}} c_{uv} e_{uv}^j \leq C_L, \quad \forall j. \tag{26}$$

If the number of available wavelength channels on each fiber is W, the following constraint (27) should be included in the ILP.

$$\sum_j e_{uv}^j + \sum_{l\in \boldsymbol{L}} w_l^{uv} \leq W, \quad \forall (u,v) \in \boldsymbol{E}. \tag{27}$$

In practice, whether we need to include one or both of the above optional constraints can be decided according to the particular engineering considerations.

V. Numerical Results

The ILP is implemented using ILOG CPLEX 11.0 [15] on a server with 3 GHz Intel Xeon CPU 5160. In our ILP, J should be set large enough to accommodate all necessary protection cycles in the solution. Otherwise, the ILP may not be able to generate a feasible solution, or a solution is generated but its optimality cannot be guaranteed. On the other hand, a large J may increase the running time of the ILP due to more variables involved. As we will show later, for a given network, the required number of CFP cycles tends to be less than that of link-based p-cycles. As a result, generally we can use the same approach as in [11] (for link-based p-cycle design without candidate cycle enumeration) to determine a suitable value of J. In this paper, we set $J = 3$ and $\lambda = 0.01$, and hop-count is used as the cost metric (i.e., $c_{uv} = 1$ for each link (u,v)). We compare CFP solutions with link-based p-cycle solutions, which are obtained

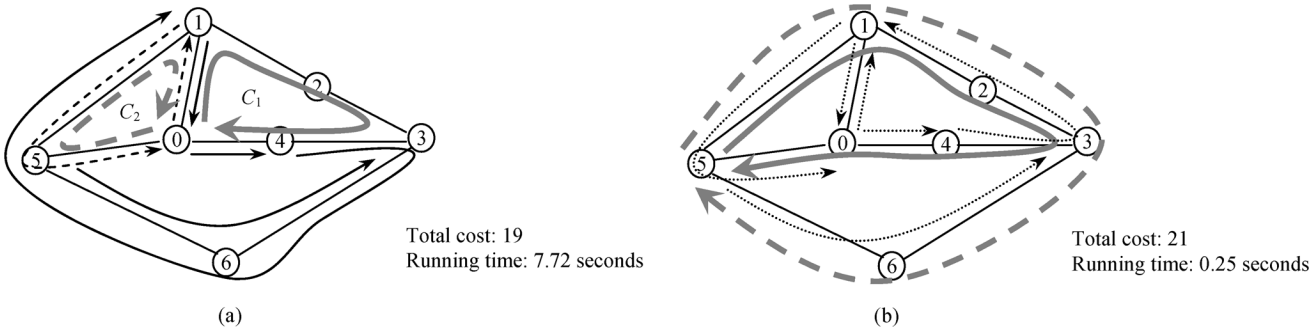

Fig. 6. A simple network with $\boldsymbol{L} = \{L_{sd}\} = \{L_{01} = 1, L_{04} = 1, L_{10} = 2, L_{41} = 1, L_{53} = 1\}$. (a) Optimal CFP solution. (b) Optimal p-cycle solution.

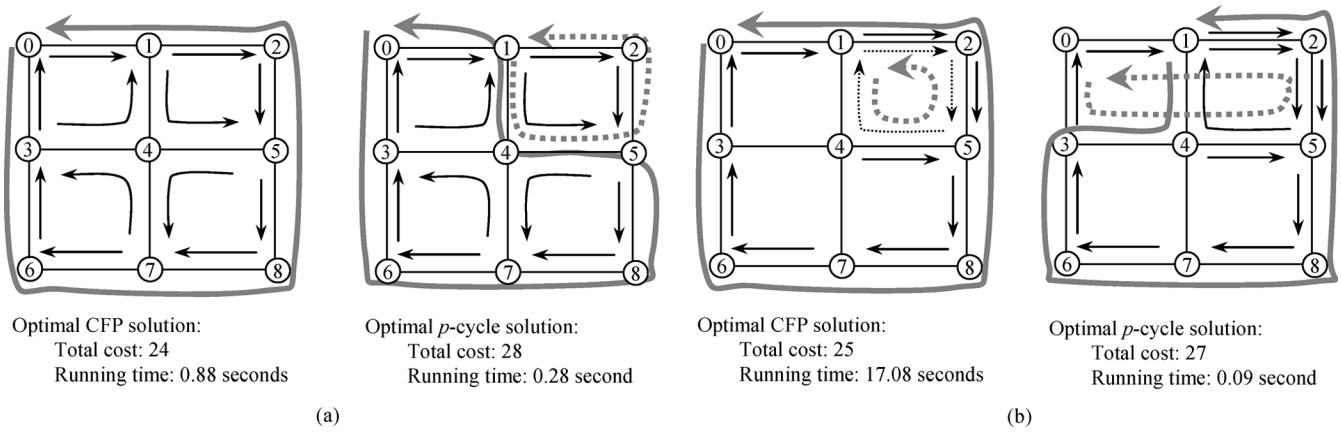

Fig. 7. CFP versus p-cycle protection in a 3 × 3 Manhatton topology. (a) $\boldsymbol{L} = \{L_{sd}\} = \{L_{01} = 1, L_{12} = 1, L_{25} = 1, L_{58} = 1, L_{87} = 1, L_{76} = 1, L_{63} = 1, L_{30} = 1, L_{31} = 1, L_{15} = 1, L_{57} = 1, L_{73} = 1\}$.. (b) $\boldsymbol{L} = \{L_{sd}\} = \{L_{01} = 1, L_{12} = 2, L_{25} = 2, L_{58} = 1, L_{87} = 1, L_{76} = 1, L_{63} = 1, L_{30} = 1, L_{51} = 1, L_{45} = 1\}$.

from a modified ILP based on [11] for directed networks. For fair comparisons, we also carry out a joint design of working and spare capacity placement in the link-based p-cycle scenario. FIPP p-cycle is not considered in the experiments because all the connections in the FIPP p-cycle scenario must be bidirectional and symmetrically loaded on the same route, which is not assumed in this study. As we have analyzed in Section III.F, CFP is more general than FIPP p-cycle with a faster optical recovery speed, and a better capacity efficiency of CFP can be proved by theoretical analysis.

We first consider a simple network with a given traffic matrix $\boldsymbol{L}$ in Fig. 6. Fig. 6(a) shows the optimal CFP solution in contrast to that in Fig. 6(b) for link-based p-cycle. We can see that the length of the CFP cycles tends to be shorter than that of the link-based p-cycles. In Fig. 6(a), two CFP cycles C_1 and C_2 protect the six lightpaths against any single link failure. The lightpaths protected by C_1 are shown by the solid arrows, and those protected by C_2 are shown by the dashed ones. We can see that both lightpaths $4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 1$ and $5 \rightarrow 6 \rightarrow 3$ can be protected by C_1 against a failure at link (5, 6), with the set of failure-aware switching nodes $\{1, 3, 5, 6\}$. Since the two lightpaths are partners of each other at this link, the backup path of one lightpath reuses the stub of the other, such that the restored traffic can be bridged onto the spare capacity on C_1. Then, the two restored lightpaths share the spare capacity on C_1 in a conflict-free manner to reach their destination nodes. Note that not only the source node 5 of lightpath $5 \rightarrow 6 \rightarrow 3$ but also the two end nodes of the failed link (5, 6) are not on cycle C_1. Such a protection is impossible in both link-based and FIPP p-cycle scenarios. On the other hand, if link (1, 5) fails, lightpaths $4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 1$ and $1 \rightarrow 5 \rightarrow 0$ will be disrupted, and the set of switching nodes is $\{0, 1, 5\}$. Nodes 1 and 5 detect an adjacent failure and node 0 detects a remote one. Stub $5 \rightarrow 0$ of $1 \rightarrow 5 \rightarrow 0$ is reused to bridge the restored traffic of $4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 1$ onto C_1 at node 0, and then the traffic goes along C_1 to reach its destination node 1. Meanwhile, lightpath $1 \rightarrow 5 \rightarrow 0$ is protected by the backup path $1 \rightarrow 0$ on C_2. Compared with the CFP solution in Fig. 6(a), the link-based p-cycle solution in Fig. 6(b) increases the total capacity by 10.53%. A careful study also shows that, the average end-to-end hop-count of the backup paths is 3.55 for CFP in Fig. 6(a), but 5.1 for the link-based p-cycle protection in Fig. 6(b).

Next, we consider the 3×3 Manhatton topology in Fig. 7, where the broad-brush arrows denote the protection cycles and other regular arrows denote the lightpaths. In Fig. 7(a), the p-cycle solution increases the total capacity by 16.67% over the CFP solution. Since the routing in both scenarios is the same, we can also compare the required spare capacity only. The p-cycle solution increases the spare capacity by 50% over CFP. Besides, only one CFP cycle is required in contrast to two p-cycles for the same traffic matrix. This example shows that, compared with the link-based p-cycle protection, CFP also tends to reduce the required number of cycles due to its

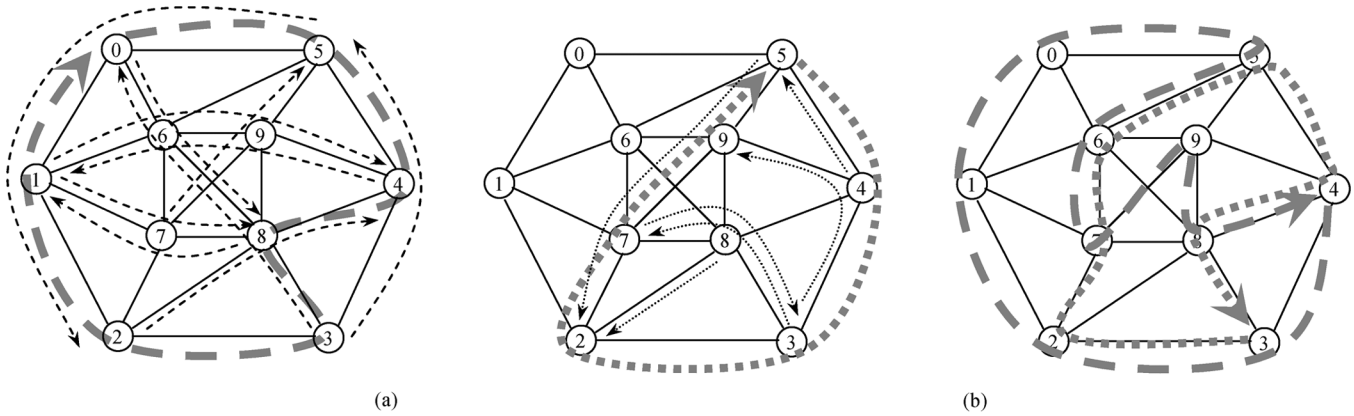

Fig. 8. An example in SmallNet with 10 nodes and 22 links. The traffic matrix includes 16 lightpaths where $\boldsymbol{L} = \{L_{sd}\} = \{L_{08} = 1, L_{14} = 1, L_{18} = 1, L_{24} = 1, L_{30} = 1, L_{35} = 1, L_{37} = 1, L_{39} = 1, L_{41} = 1, L_{45} = 1, L_{52} = 2, L_{73} = 1, L_{75} = 1, L_{81} = 1, L_{82} = 1\}$. The working paths in the p-cycle solution are the same as those in the CFP solution except for L_{41} and L_{75}, where L_{41} takes $4 \rightarrow 8 \rightarrow 6 \rightarrow 1$ and L_{75} takes $7 \rightarrow 6 \rightarrow 5$. For simplicity, working paths are not shown in the p-cycle solution. (a) Optimal CFP solution consisting of two cycles and their protected lightpaths (total cost: 48, running time: 39004.59 s). (b) Optimal p-cycle solution consisting of two cycles (total cost: 52, running time: 17.22 s).

more powerful protection capability. However, the average end-to-end hop-count of the backup paths is 6 in CFP, which is slightly larger than 5.75 in the p-cycle scenario. This is because the p-cycle solution in Fig. 7(a) uses one more (dotted) cycle which has a shorter length than the only CFP cycle. As a result, traffic protected by this cycle has a shorter backup path. For another traffic matrix in Fig. 7(b), CFP needs two cycles to protect all the twelve lightpaths, where the solid/dotted lightpaths are protected by the solid/dotted CFP cycles, respectively. We can see that lightpath $4 \rightarrow 5$ is protected by the solid CFP cycle. Upon a link failure at (4, 5), it reuses the stub $4 \rightarrow 1$ of the dotted lightpath $5 \rightarrow 4 \rightarrow 1$, which is protected by the other (dotted) CFP cycle. By comparing the two dotted cycles in Fig. 7(b), we further confirm that CFP cycles tend to be shorter in length than link-based p-cycles. In Fig. 7(b), the p-cycle solution increases 8% of the total capacity (or 16.67% spare capacity) over the CFP solution. On the other hand, the average end-to-end hop-count 5.77 of the backup paths is the same for both scenarios. We also note that in both Fig. 7(a) and (b) most of the lightpaths only take a single hop in their working path, and thus no backhaul can be observed on the corresponding backup path. When the average hop-count of the backup paths is compared among the schemes, link-based p-cycle benefits from this more than CFP, because the backhaul problem in p-cycle is more serious than that in CFP.

Finally, we consider the SmallNet topology taken from [16] as shown in Fig. 8, where the traffic matrix includes sixteen lightpaths. Fig. 8(a) shows the optimal CFP solution in contrast to Fig. 8(b) for the link-based p-cycle scenario. For clarity, in the CFP solution we separate the two cycles and the lightpaths protected by each cycle. By comparing the cycle length of both the dashed and the dotted cycles between CFP and p-cycle solutions, again we can confirm that the cycles in the proposed CFP scheme tend to have a shorter cycle length. We can see that none of the two CFP cycles traverses through node 6, which has to be traversed by both p-cycles in Fig. 8(b). In Fig. 8(a), the dashed lightpath $7 \rightarrow 9 \rightarrow 5$ can be protected by the dashed CFP cycle, because its partner at links (7, 9) and (5, 9) is the dotted lightpath $5 \rightarrow 9 \rightarrow 7 \rightarrow 2$ which is protected by the dotted cycle. Similarly, the partner of the dotted lightpath $8 \rightarrow 2$ at link (2, 8) is the dashed lightpath $2 \rightarrow 8 \rightarrow 4$. Though the two lightpaths are protected by different cycles, the former can reuse the stub released from the latter against a link failure at (2, 8). Compared with the CFP solution in Fig. 8(a), the p-cycle solution in Fig. 8(b) increases the total capacity by 8.33%, and the spare capacity by 30.77%. Besides, the average end-to-end hop-count of the backup paths is 4.89 in the CFP solution, in contrast to 7.03 in the p-cycle scenario. We also observe that the ILP running time in CFP design is much longer than that for link-based p-cycle design, and it increases rapidly with the number of lightpaths and the network size. Note that we have set $J = 3$ for all the examples in this paper. If J becomes larger, the running time required to generate an optimal CFP solution will be greatly increased. With the limitation of the currently available computation capabilities, the above observations indicate that it is quite complex to find an optimal CFP solution for a given network, though the practical deployment of CFP mechanism in real optical networks is simple after the solution is obtained.

Future research on CFP may focus on the following five aspects: 1) we have considered a joint design of CFP by allocating both working and spare capacity at the same time. Another ILP can be formulated by considering only spare capacity allocation for a given set of lightpaths which have been routed according to some routing scheme (such as shortest path routing); 2) it would be very interesting to study whether the ILP model formulated in this paper can be further simplified to render a much shorter running time; 3) since it is quite complex to find an optimal CFP solution by the ILP approach, efficient heuristics are desired. Due to the considerations on those distinct features of CFP (such as cooperative stub reuse, node-autonomy and failure-independency), we may not be able to easily find a good heuristic for CFP design; 4) the optical pre-cross-connection of spare capacity in this paper is based on simple cycles, where a simple cycle can pass through a node at most once. It would be interesting to know how much capacity gain can be further achieved by extending the pre-cross-connected structure of CFP to non-simple cycles [17] (where a non-simple cycle can pass through a node multiple times) and trails [7], [17] (which

is an acyclic structure); and 5) CFP is shown to achieve much better performance than those p-cycle based schemes in small size networks. It is important to study whether CFP can perform even better in large-size networks with heavy traffic loads. In fact, this is quite promising because the flexibility and possibility of cooperative stub reuse can be greatly boosted when network size and traffic load increase. Note that a link-based p-cycle tends to pass through or straddle as many links as it can, whereas a CFP cycle can be much shorter in length due to the unique feature of cooperative stub reuse. In a large-size network with heavy traffic load, more lightpaths can be involved in cooperative stub reuse and also the stubs become longer. Compared with link-based p-cycle, this helps to reduce the length of the CFP cycles, or the spare capacity required for full protection.

VI. Conclusion

We proposed a novel protection scheme called cooperative fast protection (CFP) in WDM networks to protect each lightpath against any single link failure. Based on the observation that a link failure can be detected not only by the two end nodes of the failed link but also by the destination nodes of all the disrupted lightpaths, CFP allows all those failure-aware nodes to carry out protection switching in a node-autonomous and failure-independent manner. Another distinct feature of CFP is that it enables cooperative stub reuse among different lightpaths, such that the backup paths can be set up using both the stubs and the pre-cross-connected spare capacity on the CFP cycles. Upon a link failure, CFP reroutes each disrupted lightpath directly to its destination node along the cycle and thus the backhaul problem can be effectively mitigated. The unique features of CFP also allow each lightpath to be properly protected even if the two end nodes of the failed link and the source node of the disrupted lightpath are not on the protecting cycle. Compared with its link-based p-cycle counterpart, a CFP solution tends to include a smaller number of cycles with shorter cycle lengths. We formulated an ILP for CFP design to jointly optimize both working and spare capacity placement. Theoretical analysis and numerical results showed that CFP significantly outperforms link-based p-cycle and FIPP p-cycle by achieving faster optical recovery with higher capacity efficiency.

References

[1] P.-H. Ho and H. T. Mouftah, "Shared protection in mesh WDM networks," *IEEE Commun. Mag.*, vol. 42, no. 1, pp. 70–76, Jan. 2004.

[2] J. Li and K. L. Yeung, "A novel two-step approach to restorable dynamic QoS routing," *J. Lightw. Technol.*, vol. 23, no. 11, pp. 3663–3670, Nov. 2005.

[3] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. IEEE ICC'98*, Jun. 1998, vol. 1, pp. 537–543.

[4] G. X. Shen and W. D. Grover, "Extending the p-cycle concept to path segment protection for span and node failure recovery," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 8, pp. 1306–1319, Oct. 2003.

[5] A. Kodian and W. D. Grover, "Failure-independent path-protecting p-cycles: Efficient and simple fully preconnected optical-path protection," *J. Lightw. Technol.*, vol. 23, no. 10, pp. 3241–3259, Oct. 2005.

[6] P.-H. Ho, J. Tapolcai, and H. T. Mouftah, "On optimal diverse routing for shared protection in mesh WDM networks," *IEEE Trans. Reliabil.*, vol. 53, no. 6, pp. 216–225, Jun. 2004.

[7] T. Y. Chow, F. Chudak, and A. M. Ffrench, "Fast optical layer mesh protection using pre-cross-connected trails," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 539–548, Jun. 2004.

[8] R. R. Iraschko, M. MacGregor, and W. D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh survivable networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 325–336, Jun. 1998.

[9] D. A. Schupke, C. G. Gruber, and A. Autenrieth, "Optimal configuration of p-cycles in WDM network," in *Proc. IEEE ICC'02*, May 2002, vol. 5, pp. 2761–2765.

[10] D. A. Schupke, "Analysis of p-cycle capacity in WDM networks," *Photonic Network Commun.*, vol. 12, no. 1, pp. 41–51, Jul. 2006.

[11] B. Wu, K. L. Yeung, and P.-H. Ho, "ILP formulations for p-cycle design without candidate cycle enumeration," *IEEE/ACM Trans. Netw.* [Online]. Available: http://www.eee.hku.hk/research/doc/tr/TR2008001_IFDCC.pdf

[12] J. Shin, Y. Kwon, and J. Ko, "Optical supervisory channel subsystem for 1.6 T WDM transmission system," in *Proc. 6th Int. Conf. Adv. Commun. Technol.*, Feb. 2004, vol. 1, pp. 402–404.

[13] B. Wu, P.-H. Ho, and K. L. Yeung, "Monitoring trail: On fast link failure localization in WDM mesh networks," *J. Lightw. Technol.*, vol. 27, no. 18, pp. 4175–4185, Sep. 2009.

[14] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[15] [Online]. Available: www.ilog.com

[16] H. Zeng, C. Huang, and A. Vukovic, "A novel fault detection and localization scheme for mesh all-optical networks based on monitoring-cycles," *Photonic Network Commun.*, vol. 11, no. 3, pp. 277–286, May 2006.

[17] B. Wu, K. L. Yeung, and P.-H. Ho, "ILP formulations for non-simple p-cycle and p-trail design in WDM mesh networks," *Elsevier Computer Networks*, to be published.

Bin Wu (S'04–M'07) received the B.Eng. degree from Zhe Jiang University, Hangzhou, China, in 1993, the M.Eng. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 1996, and the Ph.D. degree from the University of Hong Kong, Hong Kong, in 2007.

During 1997–2001, he served as the department manager of TI-Huawei DSP Co-lab, Huawei Tech. Company Ltd, Shenzhen, China. Currently, he is a Postdoctoral Research Fellow with the University of Waterloo, Waterloo, ON, Canada.

Pin-Han Ho received the B.Sc. and M.Sc. degrees from National Taiwan University, Taipei, in 1993 and 1995, respectively, and the Ph.D. degree from Queen's University, Kingston, ON, Canada, in 2002. His dissertation focused on optical communications systems, survivable networking, and QoS routing problems.

He then joined the Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, Canada, as an Assistant Professor in the same year. He is the author/coauthor of more than 100 refereed technical papers and book chapters and the coauthor of a book on optical networking and survivability.

Dr. Ho was the recipient of the Distinguished Research Excellence Award in the ECE Department at the University of Waterloo, the Early Researcher Award in 2005, the Best Paper Award at SPECTS'02 and the ICC'05 Optical Networking Symposium, and the Outstanding Paper Award in HPSR'02.

Kwan L. Yeung (S'93–M'95–SM'00) was born in 1969. He received the B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong in 1992 and 1995, respectively.

He joined the Department of Electrical and Electronic Engineering, The University of Hong Kong, in July 2000, where he is currently an Associate Professor and the Information Engineering Program Co-Director. Before that, he spent five years with the Department of Electronic Engineering, City University of Hong Kong, as an Assistant Professor. During the summer of 1993, he served with the Performance Analysis Department, AT&T Bell Laboratories (now Bell Labs, Lucent Technologies), Holmdel, NJ, as a Member of Technical Staff. His research interests include next-generation Internet, packet switch/router design, all-optical networks and wireless data networks. He holds two patents and has published over 120 papers in international journals and conferences since 1993.

János Tapolcai (M'09) received the M.Sc. degree in technical informatics and the Ph.D. degree in computer science from Budapest University of Technology and Economics (BME), Budapest, Hungary, in 2000 and 2005, respectively.

Currently, he is an Associate Professor with the High-Speed Networks Laboratory at the Department of Telecommunications and Media Informatics at BME. His research interests include applied mathematics, combinatorial optimization, linear programming, linear algebra, routing in circuit switched survivable networks, availability analysis, grid networks, and distributed computing. He has been involved in a few related European and Canadian projects (IP NOBEL; NoE e-Photon/ONe; BUL). He is an author of over 30 scientific publications.

Dr. Tapolcai was the recipient of the Best Paper Award at ICC'06.

Hussein T. Mouftah (F'90) joined the School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada, in September 2002 as a Canada Research Chair Professor (Tier 1). He has been with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada, since 1979, where prior to his departure in August 2002 he was a full Professor and department associate head, after three years of industrial experience mainly at Bell Northern Research of Ottawa (now Nortel Networks). He has spent three sabbatical years at Nortel (1986–1987, 1993–1994, and 2000–2001) conducting research in the areas of broadband packet switching networks, mobile wireless networks, and quality of service over the optical Internet. He is the author or coauthor of six books and more than 850 technical papers and 10 patents.

Dr. Mouftah is a Fellow of the Canadian Academy of Engineering (2003), Fellow of the Engineering Institute of Canada (2005) and Fellow RSC: The Academies of Canada (2008). He served as Editor-in-Chief of the *IEEE Communications Magazine* (1995–1997), IEEE Communications Society Director of Magazines (1998–1999), Chair of the Awards Committee (2002–2003), Director of Education (2006–2007), and Member of the Board of Governors (1997–1999 and 2006–2007). He is also the founding Chair of two of IEEE Communications Society Technical Committees (TCs): Optical Networking TC (2002–2004) and Ad Hoc and Sensor Networks TC (2005–2007). He has been a Distinguished Speaker of the IEEE Communications Society (2000–2007). He was the recipient of the 1989 Engineering Medal for Research and Development of the Association of Professional Engineers of Ontario (PEO). He has also received eight Outstanding/Best Paper Awards, the IEEE Canada Outstanding Service Award (1995), and the CSIM Distinguished Service Award of the IEEE Communications Society (2006). In 2004 Dr. Mouftah received the IEEE Communications Society Edwin Howard Armstrong Achievement Award and the George S. Glinski Award for Excellence in Research from the Faculty of Engineering, University of Ottawa. In 2006, he was honoured with the IEEE McNaughton Gold Medal and the Engineering Institute of Canada Julian Smith Medal. In 2007 he was the recipient of the Royal Society of Canada (RSC) Thomas W. Eadie Medal. Most recently, he received the University of Ottawa 2007–2008 Award for Excellence in Research and the ORION Leadership Award of Merit (2008).